



UiT The Arctic University of Norway

Faculty of Electrical Engineering

Impact on Distribution Network Hosting Capacity using Machine Learning Based Reactive Power Support from PV Smart Inverters

Sepehr Safavi

Master's thesis in Electrical Engineering ELE-3900 May 2024



Table of Contents

Abbreviation.....	8
1 Introduction	1
1.1 Background and Motivation	1
1.2 Problem Statement	1
1.3 Research Objectives	3
1.4 Thesis Limitation.....	3
1.5 Scope of the Thesis.....	3
2 Paper review	4
2.1 Smart Distribution Network.....	4
2.2 Voltage problems	4
2.2.1 Overloading and power loss problems	5
2.2.2 power quality problems	5
2.2.3 protection problems.....	5
2.3 Hosting capacity: enhancement techniques	6
2.3.1 Reactive power control.....	7
2.4 Machine learning background.....	8
2.4.1 Reinforcement Learning	8
2.4.2 Unsupervised learning	9
2.4.3 Supervised Learning	9
2.5 ML Method Selection.....	9
2.6 SUPPORT VECTOR MODEL (SVM)	10
2.7 SVM with RBF Kernel for Reactive Power Prediction.....	12
2.7.1 Why RBF Kernel SVM is a Good Choice.....	12
2.8 Tuning Gamma (γ) and Sigma (σ) in RBF SVM.....	12
3 Methodology	14
3.1 Flow chart	14

3.2	tools used	15
3.2.1	CIGRE Network.....	16
4	Method implementation	18
4.1	power flow analysis.....	18
4.2	DATASET FOR MODELING	20
4.2.1	SAMPLE RESULTS	20
5	Results.....	23
5.1	SVM model.....	23
5.2	SVM Analysis	25
5.2.1	Underfitting and Overfitting	25
5.2.2	Accuracy	26
5.2.3	Robustness	26
5.2.4	Performance	26
6	Conclusion and Future Research	27
	Appendix A.....	28
	Appendix B.....	34
	Appendix C.....	37
	References.....	42

List of Tables

List of Figures

Figure 1.1	Unilateral vs distributed power system.....	2
Figure 2	Factors affect HC.....	6
Figure 3	Reactive power control curve.....	7
Figure 4	Basic framework of reinforcement learning	9
Figure 5,	Flow Chart.....	14
Figure 6,	CIGRE Medium Voltage Distribution Network [19]	16
Figure 7,	CIGRE Model	17

Figure 8 voltage bus 19

Figure 9, reactive power bus 19

Figure 10, reactive power sample buses 21

Figure 11, voltage sample buses 22

Figure 12, ML Predic Bus2..... 23

Figure 13, ML Predic Bus3..... 24

Figure 14, ML Predic Bus4..... 24

Figure 15, ML Predic Bus5..... 25

Abstract

Nowadays, due to growth of advanced technology and its impact on people lives, there is an increasing need for aligning and optimizing infrastructure with people needs. Harnessing renewable energy from different sources, store and eventually transmit and distributed it is an all-time challenge which nowadays by implementing and proper maintenance we lead to having higher efficiency to the point where we are able to maximize usage of these resources.

There is a concept called Hosting capacity to measure proper amount of integration of distribution generators into the grid without causing any malfunctioning in the system. The goal is to find a way to enhance the HC.

Predicting reactive power is a good solution to halt voltage violation. Techniques such as optimal power flow has been recommended for that however, machine learning algorithms is new approach for reactive power predication due to its better performance and ability to consider dominant variables affected on the data set.

Thus, the literature review has been conducted to choose a ML algorithm for time series prediction of reactive power on a chosen Network. After, the methodology has been illustrated. Then, a dataset has been generated by doing power flow analysis and used in the ML algorithm in order to compare the results. In this case, for some parts the results were not fulfilling compared to the generated data where the affected factors have been discussed and future works proposed.

Abbreviation

HC Hosting Capacity

ML Machine Learning

OPF Optimal Power Flow

SVM Support Vector Machine

PV Photovoltaic

HV High Voltage

LV Low Voltage

SDN Smart Distribution Network

AMI Advanced metering infrastructure

ICT information and communication technologies

RNNs Recurrent Neural Networks

GBMs Gradient Boosting Machines

LSTM Long-Short Term Memory

CNN Convolutional Neural Network

1 Introduction

In this section the background and problem will be discussed along with the thesis scope.

1.1 Background and Motivation

The global shift towards renewable energy is accelerating due to advancements in technology, economic incentives, environmental considerations, and political and social support. However, integrating high levels of distributed generation (DG) into power grids can introduce several challenges. These include voltage fluctuations, increased line losses, transformer and feeder overloads, malfunctions in protective devices, and elevated harmonic distortions that can breach the thresholds set by international norms. Such issues arise when the grid's hosting capacity (HC) is exceeded. Thus, evaluating and improving the hosting capacity of distribution networks is becoming a critical priority for both utility companies and DG developers, framing it as a key strategy for ensuring the smooth integration of renewable energy sources.[1]

More importantly, the rise of artificial intelligence and its affect on technology is a proper case to study which in this thesis it will be used. Meaning, using ML algorithms to facilitate and tackle current issues that are exist, particularly in power system.

1.2 Problem Statement

Historically, power distribution systems operated on a simple principle where electricity flowed in one direction, from centralized generation facilities through the transmission and distribution lines to consumers. The introduction and expansion of distributed generation (DG) technologies, such as solar photovoltaics (PV) and wind turbines, have revolutionized this traditional flow, creating a more complex network where power can originate from multiple sources. [2] This transformation has been propelled by a mix of political, social, economic, environmental, and technical factors. Despite the benefits, the surge in DG adoption can have detrimental effects on the grid, including potential overvoltage issues, the thermal overloading of network components, an increased likelihood of surpassing the short circuit capacities of equipment, and the incorrect functioning of protective devices, all of which could compromise system integrity and reliability.[3]

Differences between hierarchic and distributed (smart) grid

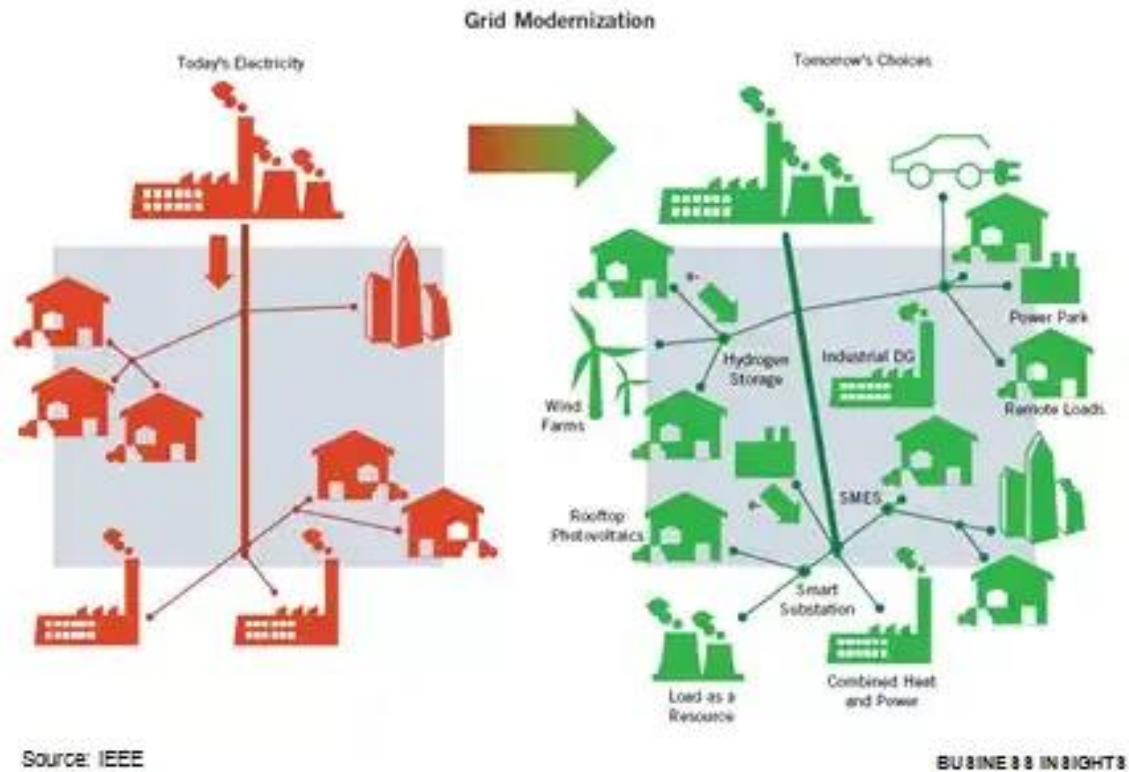


Figure 1.1 Unilateral vs distributed power system

In the context of deregulated energy markets, there is a conflict between owners or investors of distributed generation (DG) and distribution system operators (DSOs). DG investors are eager to increase the integration of DG units into the electrical grid, whereas DSOs are wary of the complications that may arise from too much DG integration. To mediate this disparity and make objective decisions on whether to approve or deny new DG integration proposals, the concept of hosting capacity (HC) was introduced as a fair and transparent mechanism.

This strategy focuses on understanding the technical limits set by both the operators of the system and the customers. It defines hosting capacity (HC) as the highest amount of distributed energy that can be added to the power system without causing problems. The calculation of HC isn't straightforward or uniform; it varies and should be done considering different factors like changes in voltage and frequency, the risk of equipment overheating, power quality issues, and protection system failures. [4]

Using ML in reactive power prediction is a proper tool due to its nature. Meaning, having access to a big amount of real time data to sort out the best value among them in a decentralized network will be a good environment for ML to be implemented.

In order to get the maximum results out of it this needs to be tested in a smaller case network to see the reliability of the algorithms and accordingly it could be implemented in larger scale networks.

1.3 Research Objectives

This study covers following:

- Factors affected HC
- ML approach
- Time series forecasting reactive power
- Analysis of the results

1.4 Thesis Limitation

- A small network has been chosen, considering 4 buses
- A limited dataset has been used for analysis
- ML algorithm has been used on a time series of generated dataset

1.5 Scope of the Thesis

Scope of the thesis will be included in the following sections

- **Introduction:** in this section background and the problem have been discussed along with objectives and limitations of thesis.
- **Paper Review:** it covers dominant factors affecting HC and ML approaches to tackle the reactive power prediction.
- **Methodology:** in this section the overview of how to approach to tackle the problem by showing a flow chart and the tools that have been used is discussed.
- **Implementation:** this part covers the dataset generation and implementing the ML algorithm
- **Results:** the results will be discussed here. ML results will be analyzed in terms of accuracy, robustness and performance.
- **Conclusion:** overall conclusion and future work are mentioned in this section.

2 Paper review

2.1 Smart Distribution Network

Traditional electricity grids are one-way streets: power plants generate electricity that flows through passive infrastructure to end users. Smart network distribution (SND) disrupts this model by introducing intelligence and two-way communication into the grid. This transformation offers a range of benefits, including improved efficiency, reliability, and integration of renewable energy sources.

The foundation of SND lies in the convergence of information and communication technologies (ICT) with the power grid. Advanced metering infrastructure (AMI) replaces traditional meters with intelligent devices that collect real-time data on energy consumption and grid conditions.

This data empowers utilities to monitor and manage the grid proactively, identifying potential problems and optimizing power flows.[42]

The two-way communication enabled by SND unlocks further possibilities. Distributed generation (DG), such as solar panels, can be integrated more effectively. Smart grids can also facilitate demand response programs, where consumers are incentivized to adjust their electricity usage during peak hours, alleviating strain on the grid.[43]

As mentioned in introduction, HC is the main measure to evaluate the proper integration of Der through grid where needs to be improved; however, there are some limitations which need to be tackle including:

2.2 Voltage problems

When a distributed generation (DG) unit generates more power than what is needed for the load, the excess electricity is fed back into the grid. This can lead to an increase in voltage at the point where the load is connected, potentially causing the feeder close to it to become overloaded. The primary issues that exceed acceptable thresholds due to high levels of DG integration are the rise in voltage and the surpassing of the maximum continuous current that the wires (conductor ampacity) can handle. [5]

The effects of DG integration on the network's voltage profile and the voltage increase (DV_{rise}) across various bus locations were estimated. [6]

$$\Delta V_{rise} \cong \frac{P \cdot R + Q \cdot X}{|V_n|}$$

Where V_n is a system parameter specified by the DSO and can be adjusted by controlling the set-point of the OLTC in the upstream transformer.

The resistance (R) and reactance (X) are determined by the specific attributes of the system. To mitigate voltage increases, enhancing the system's resistance and reactance through cable upgrades is a strategy. Nonetheless, this approach may not always be feasible due to financial constraints and the practical challenges of updating existing cables in densely populated urban

areas. The (X/R) ratio remains a fixed value, reflective of the distinct features of the distribution system. A higher (X/R) ratio indicates that the (Q*X) term is more significant. Therefore, managing the system's reactive power (Q) emerges as a viable strategy to prevent overvoltage situations. Problems with overvoltage and rapid voltage fluctuations become significantly apparent when numerous photovoltaic (PV) systems are installed close to the end of long and lightly loaded feeders. [7]

2.2.1 Overloading and power loss problems

Optimal placement of distributed generation (DG) units can cut down on losses within the feeder and alleviate the load on the supply transformers. Typically, the process of minimizing feeder losses through the integration of DGs is framed as an optimization challenge. In practical terms, the decision on where to place DG units and how powerful they should be is often made by the DG investor or owner. The most advantageous condition is achieved when the DG is situated near the load that is being served.[8]

The integration (DG) units can thus alleviate the risk of overloading, decrease the losses within the system, extend the lifespan of equipment, and enhance the thermal performance of both feeders and transformers. [9]

The worst-case scenario which leads to the greatest risk of overloading occurs with the maximum generation and minimum loading. If the DG output power is higher than the local load power, i.e. $P_{DG} > P_{load}$; the DG starts to inject power to the upstream network after meet the load demand. Therefore, reverse power flow will arise and may lead to exceeding the thermal capacity of the equipment (transformers and feeders). [10]

2.2.2 power quality problems

Power quality focuses on the relationship between the electrical utility and its consumers, specifically regarding how voltage and current disturbances. deviations from the ideal sinusoidal waveform affect both the network and its users. The goal of voltage quality is to ensure that the voltage supplied to customers is within acceptable range. The excessive DG penetration may lead to some power quality disturbances such as power system harmonics, steady state voltage variations, rapid voltage changes, voltage dips and flickers. [11]

2.2.3 protection problems

When the power generated by distributed generation (DG) surpasses local demand, it can lead to reverse power flow and issues with the protection systems. High levels of DG can alter both the amount and direction of fault currents, potentially leading to protection system malfunctions, including inappropriate activations under normal conditions or failures to activate during faults. adjustments to current protection mechanisms might be necessary to accommodate the influence of DG.

Additionally, when evaluating the hosting capacity (HC) within the context of protection limitations, the analysis should consider various thresholds of DG integration. These include:

- The degree of DG penetration at which altering the settings for current or delay on any current protection relays is required.
- The point of DG integration at which the addition of extra circuit breakers or protective components to the network is recommended.
- The level of DG penetration that necessitates the replacement of an existing protection relay with a newer model. [12]

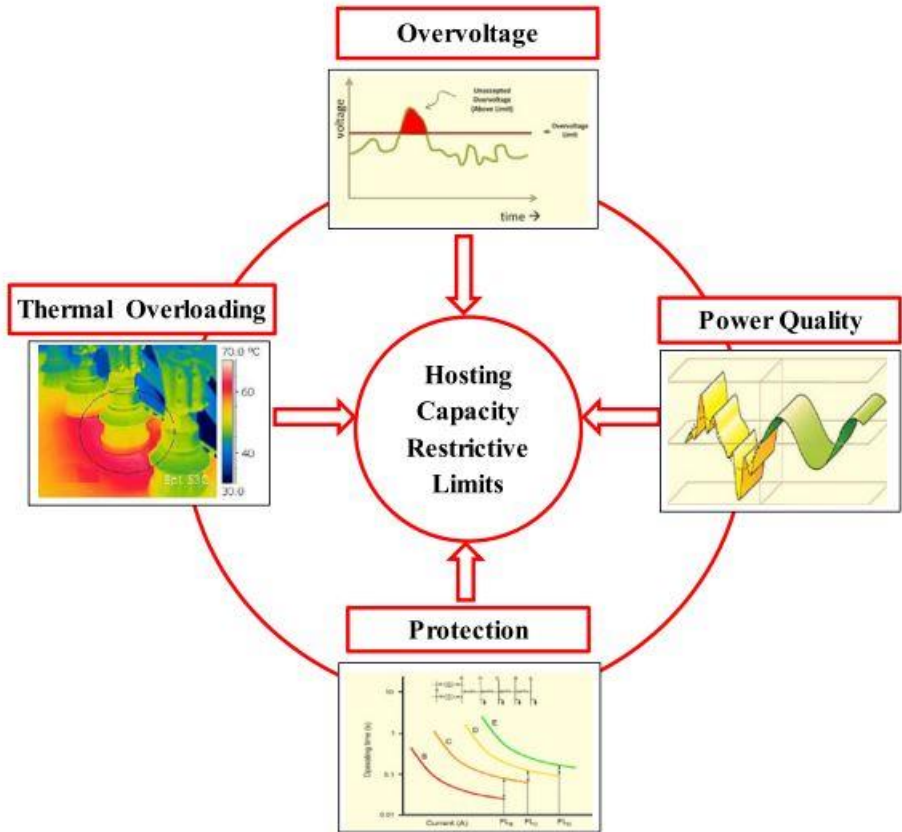


Figure 2 Factors affect HC

2.3 Hosting capacity: enhancement techniques

Currently, improving the system's hosting capacity (HC) is considered crucial for distribution system operators (DSOs). The technical strategies available for boosting the system's HC are divided into three main groups: DSO solutions, prosumer solutions and interactive solutions. [13]

Among different enhancement strategies, here we discuss particularly for reactive power control method.

2.3.1 Reactive power control

There is a relationship between the reactive power, whether injected or absorbed, in a PV inverter and the bus voltage when connected to the PV system in a grid. In one scenario which is under-voltage, the control functions in a capacitive manner, injecting reactive power until the voltage approaches the reference voltage. This control method is also utilized to increase the active power injection into the grid during under-voltage scenarios, thereby improving the HC of the system. On the other hand, when over-voltage occurs, the control operates in an inductive manner, absorbing reactive power to maintain the voltage at the bus equal to the reference value. Furthermore, it ensures that the actual power production from the PV system remains constant.[14]

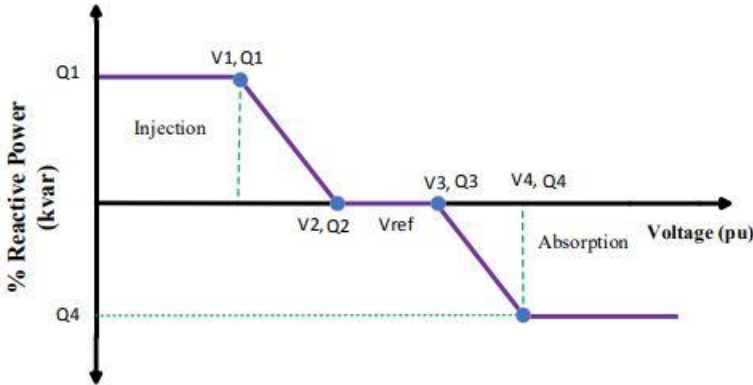


Figure 3 Reactive power control curve

Voltage rise problems are considered as the important factor for high DG penetration. Reactive power control techniques are the most effective methods for tackling over voltage problems with DSOs and end-users. Various reactive power control techniques can be utilized such as shunt and series capacitor banks, Static VAR compensator, Static compensator and DG units controlled by smart inverters.

2.3.1.1 capacitor bank

Power capacitors can be utilized in either series or parallel (shunt) configurations. In the case of series capacitors, they are typically used in high-voltage (HV) transmission lines for series compensation, enhancing the lines' capacity to transmit power. On the other hand, shunt capacitors are integrated into electrical systems to perform various functions.[15] Shunt capacitors work by modifying the power factor of the load, improving efficiency. In contrast, series capacitors by counteracting the inductive reactance in the circuit they are connected to imposes their affect.

2.3.1.2 static VAR compensator

Static var compensators regulate voltage by controlling the reactive power absorbed from or injected into the power system. It generates reactive power by switching capacitor banks when the system voltage is low, or the load is inductive. Therefore, the reactive power demand of the lagging load is provided by the static var compensator, which relieves the transmission of the distribution line. Therefore, the voltage drop decreases and the voltage at the load terminal should increase. Likewise, when the system voltage is high or the load is capacitive, the static var compensator absorbs reactive power. In this case, the static var compensator uses a reactor to dissipate the VAR in the system, thereby reducing the system voltage.

2.4 Machine learning background

In general, Machine learning involves the process of learning where data or signals are transformed into information, with the aim of extracting knowledge. This knowledge, which may consist of new data or insights about the learning system itself, can assist in making decisions or predictions.[16][17]

In the Machine Learning (ML) area, there are three main categories: unsupervised learning, supervised learning, and reinforcement learning (RL).

SVM algorithm will be chosen for this study.

2.4.1 Reinforcement Learning

Reinforcement learning is a technique in machine learning where training involves rewarding positive behavior and penalizing negative behavior. The agent, or the entity being trained, has the capability to observe and understand its surroundings, make decisions, and learn via a process of trial and error.

The main elements on RL are the agent which interacts with its environment, on each time step, the agent receives the environment state St , in which $St \in S$, means the set of all possible states, based on that the agent “acts” selecting a specific Action $At \in A(St)$. As a consequence of this action, the environment sends a reward to the agent, and the main goal for the agent is to maximize total reward overall.[18] This maximization is called Value function and the rule that the agent follows to achieve more rewards is called policy which makes a mapping of probabilities distributions of each and one of the possibilities of actions that can be executed.[19]

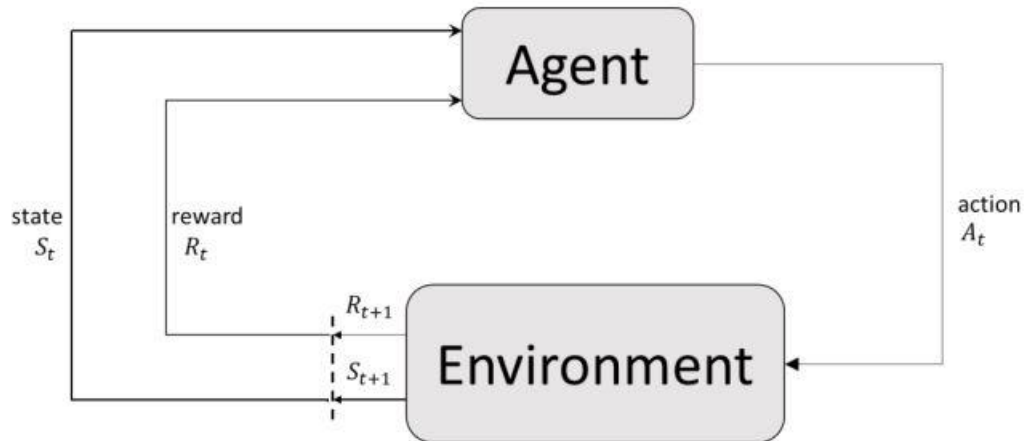


Figure 4 Basic framework of reinforcement learning

2.4.2 Unsupervised learning

Unsupervised learning is a fundamental technique in machine learning where algorithms discover hidden patterns and structures within unlabeled data. Unlike supervised learning, which involves training a model with labeled data (data where the desired output is already known), unsupervised learning operates on data without any predefined categories or labels. This allows the model to identify interesting and potentially useful relationships within the data on its own.[41]

2.4.3 Supervised Learning

In supervised learning, the algorithm is trained on a labeled dataset. This means the data includes both input features and the desired output (labels). The algorithm learns the relationship between the features and the labels, and then uses this knowledge to make predictions for new, unseen data.

2.5 ML Method Selection

Predicting the reactive power of Distributed Energy Resources (DERs) connected to a Smart Distribution Network (SDN) is crucial for maintaining grid stability and efficiency. Supervised learning is good at tasks where labeled data is readily available. In the context of SDN with DERs, historical data on past power generation and consumption patterns of the

DERs can be collected. This data, along with environmental factors influencing their output, solar irradiance for solar panels, wind speed for wind turbines, serves as labeled data for training. The labeled data allows the supervised learning model to learn the relationships between these factors and the corresponding reactive power output of the DERs. This knowledge helps the model to predict reactive power for unseen scenarios.[44]

On the other hand, Reinforcement learning is good in scenarios where the environment is complex, dynamic, and the desired outcome might not be defined. The core concept involves an agent interacting with the environment, taking actions, and receiving rewards or penalties (reinforcement signals) based on the outcomes. Over time, the agent learns to make better decisions that maximize its rewards.[45]

Overall, supervised learning offers several advantages for reactive power prediction in SDN due to the facts:

interpretability: Supervised learning models often provide better interpretability compared to reinforcement learning models. This allows us to understand the key factors influencing reactive power output, which is valuable for optimizing DER operation and control strategies within the network.

Focus on Prediction: Supervised learning is specifically designed for prediction tasks, making it a more direct approach for this application.

2.6 SUPPORT VECTOR MODEL (SVM)

Support Vector Machines (SVMs) are powerful supervised learning algorithms widely used in power system applications. They are good at finding optimal decision boundaries (hyperplanes) in high-dimensional spaces, separating data points belonging to different classes with the largest margin. This margin signifies the distance between the hyperplane and the closest data points (support vectors) from each class, ensuring good generalization for unseen data prediction.[20]

2.6.1.1 Core Principle:

SVMs aim to maximize the margin between the hyperplane and the support vectors. This ensures a clear separation between classes, leading to robust models capable of handling unseen data effectively.[20]

Mathematically, SVMs formulate the learning problem as a convex optimization problem. This formulation guarantees a unique solution and facilitates efficient computational methods.[21]

2.6.1.2 Types of SVMs:

Linear SVMs:

Suitable for data that is linearly separable in the feature space. The decision boundary is a straight line (in 2D) or hyperplane (in higher dimensions).[22]

Non-Linear SVMs:

Used for data with non-linear relationships. They use kernel functions to map the data points into a higher-dimensional space where linear separation becomes possible. Common kernels include linear, polynomial, and radial basis function (RBF).[22]

2.6.1.3 Application to Reactive Power Prediction:

SVMs can be a valuable tool for predicting reactive power values in time series data. Due to the facts

2.6.1.3.1 Data Preparation:

Historical reactive power measurements along with relevant influencing factors such as voltage, active power, time of day, weather conditions) form the training data.[23]

In this study these data have been driven by running power flow analysis considering 288 samples.

Feature engineering techniques might be necessary to create informative features from the raw data that enhance the model's learning process.[24]

2.6.1.3.2 Model Training:

Choosing an appropriate SVM type (linear or non-linear) based on the complexity of the relationship between reactive power and its influencing factors should be done. [25]

Non-linear SVMs with RBF kernels are often preferred for capturing complex non-linear patterns in power system data.

Train the SVM model on the prepared historical data. Hyperparameter tuning might be required to optimize model performance for specific dataset. Common hyperparameters for SVMs include kernel parameters and regularization coefficients.[23]

2.6.1.4 Advantages of Using SVMs for Reactive Power Prediction:

- **Effective in High-Dimensional Spaces:** SVMs can handle data with multiple influencing factors, making them suitable for complex real-world power system scenarios.[20]
- **Good Generalization:** By maximizing the margin, SVMs tend to avoid overfitting and perform well on unseen data, leading to reliable predictions.[26]

Considerations:

- **Hyperparameter Tuning:** Finding the optimal hyperparameters for the SVM model can require experimentation and expertise. Techniques like grid search or randomized search can be employed to efficiently identify suitable hyperparameter combinations.[21]

2.7 SVM with RBF Kernel for Reactive Power Prediction

It has been used other algorithms for predicting reactive power values of buses such as LSTM, CNN however, a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel is a good approach due to several factors:

Number of Samples: considering not a huge amount of data, an RBF kernel can work effectively with this sample size. Compared to more complex models like deep neural networks, SVMs can often achieve good results with smaller datasets [27].

Interpretability: SVMs offer a level of interpretability that LSTMs generally lack. The decision boundary learned by an SVM-RBF can be analyzed to understand which factors, weather conditions, historical power generation patterns, most significantly influence the reactive power output of DERs. This interpretability can be valuable for improving DER operation strategies and control mechanisms within the SDN.[46]

Computational Efficiency: Training an SVM-RBF model is typically less computationally expensive compared to training an LSTM network. This can be an advantage in resource-constrained environments where computational power might be limited.[46]

Gamma (γ) and Sigma (σ): These hyperparameters are specific to the RBF kernel and control the smoothness and flexibility of the decision boundary. Tuning these parameters is crucial for optimal model performance.[28]

2.7.1 Why RBF Kernel SVM is a Good Choice

Non-Linearity: Power system data often shows non-linear relationships between reactive power and its influencing factors (voltage, active power, etc.). RBF kernels are good at capturing such non-linear patterns by mapping data points into a higher-dimensional space where linear separation becomes possible [29].

Effectiveness: Studies have demonstrated the effectiveness of RBF SVMs for short-term reactive power forecasting, achieving good prediction accuracy.[30]

Hyperparameter Tuning: The ability to tune gamma (γ) and sigma (σ) allows us to adjust the model's complexity based on the specific characteristics of reactive power data [31].

2.8 Tuning Gamma (γ) and Sigma (σ) in RBF SVM

Gamma (γ) and Sigma (σ) are crucial hyperparameters in an SVM with an RBF kernel used for predicting reactive power values. Tuning these parameters significantly impacts the

model's ability to capture the non-linear relationships in the dataset. Here's how these factors can effectively be tuned:

Understanding the Roles:

- **Gamma (γ):** This hyperparameter controls the influence of each data point on the decision boundary.
 - **Higher Gamma (γ):** Leads to a more flexible (complex) boundary that can capture intricate non-linear patterns. However, it can also lead to overfitting if not chosen carefully. [32]
 - **Lower Gamma (γ):** Results in a smoother (less complex) boundary that might miss subtle non-linear relationships in the data. [32]
- **Sigma (σ):** This hyperparameter controls the width of the RBF function, affecting the decision boundary's smoothness.
 - **Smaller Sigma (σ):** Creates a narrower function, making the model more sensitive to local variations in the data. This can be beneficial for capturing specific patterns but might also increase computational cost and risk of overfitting. [33]
 - **Larger Sigma (σ):** Leads to a wider function, providing a smoother decision boundary that might be less sensitive to local variations but could miss some non-linear trends. [34]

Tuning Techniques:

There are common techniques for tuning gamma and sigma to find the optimal combination for reactive power prediction task:

- **Grid Search:** This method systematically evaluates a predefined grid of gamma and sigma values. We can train the SVM model with each combination and choose the one that yields the best performance metric, mean squared error (MSE) or mean absolute error (MAE) for regression tasks. However, grid search can be computationally expensive, especially for large grids. [35]
- **Randomized Search:** This technique samples random combinations of gamma and sigma values from a specified range. It can be computationally more efficient than grid search while still effectively exploring the hyperparameter space. [35]
- **Bayesian Optimization:** This advanced technique uses a probabilistic model to iteratively select promising hyperparameter combinations based on past evaluations. It can be particularly efficient for finding optimal values when dealing with expensive-to-evaluate models. [35]

3 Methodology

In this section the way to approach the algorithm will be illustrated.

3.1 Flow chart

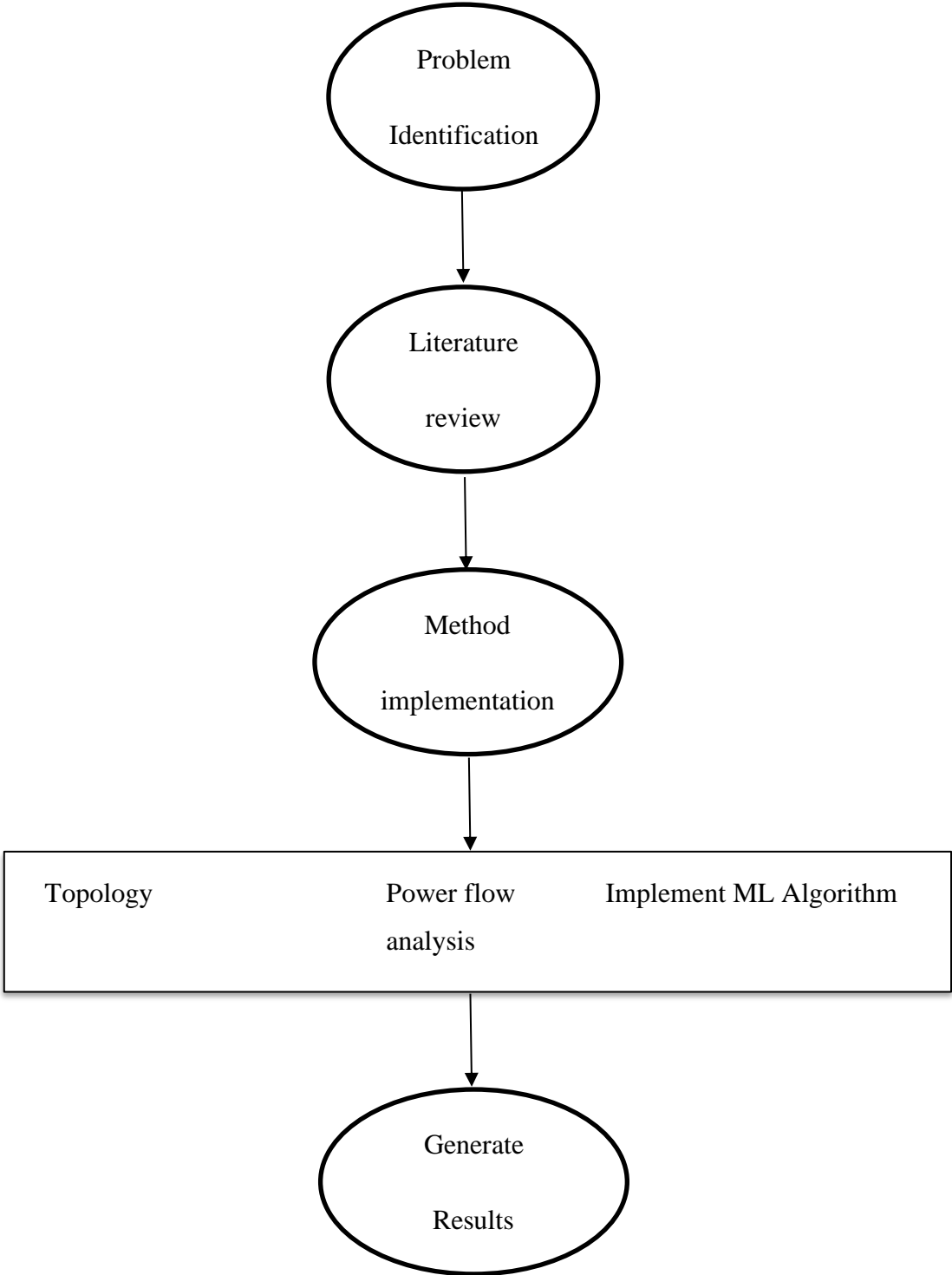


Figure 5, Flow Chart

Here as it is showed in the flow chart, an overview of methodology has been illustrated.

It starts with identify the problem and goes for reviewing papers. Then ML algorithm has been chosen and after that the proper tool to implement that will be mentioned. Later on, Power flow analysis and time series have been conducted to generate the dataset. Then the ML algorithm has been conducted to the dataset and generated the results. At the end the results have been analyzed

3.2 tools used

In this section, the tools that have been used to approach the method will be introduced, including the data, software, algorithm.

In order to implement the code for optimal power flow analysis and SVM model Anaconda Navigator with Jupyter Notebook has been used.

The data have been added to the Jupyter Notebook Environment using panda power library for generating the results.

3.2.1 CIGRE Network

Below topology has been chosen for the study. Consider DERs connected to the buses 2, 3, 4 and 5.

This Network has been chosen from sample CIGRE Networks with provided values where can be found in Appendix A.

One transformer connecting the 110kv bus bar to the 20kv bus bar.

4 DERs Connected to the bus bars

In the next sections collecting of dataset and traing model will be conducted, respectively.

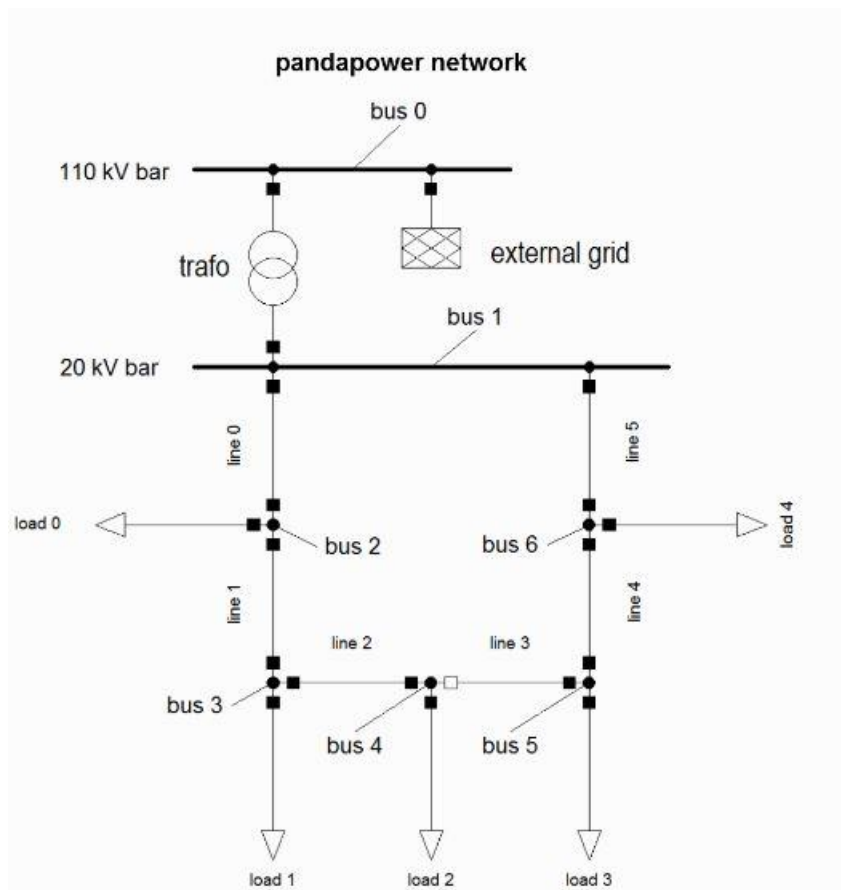


Figure 6, CIGRE Medium Voltage Distribution Network [19]

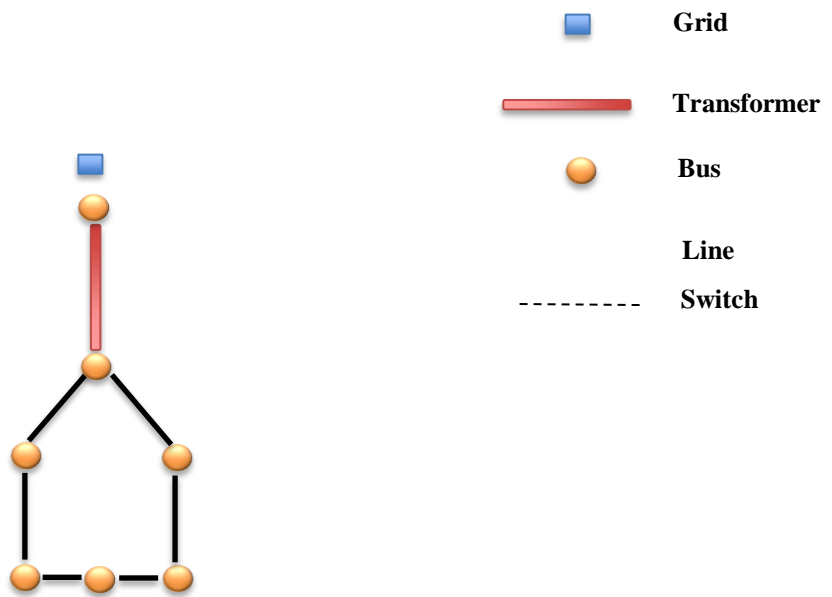


Figure 7, CIGRE Model

4 Method implementation

The method implements in this way. First. The target values for the study which here are reactive power will be generated by doing optimal power flow with 288 samples. Considering 5 minutes interval in a day. Then these samples will be tested an trained under SVM model to see if the predicted values are align with actual values, consider the generated values.

4.1 power flow analysis

Power flow analysis (PFA) is a computational method that determines the voltage magnitudes and phase angles at each bus (electrical node) in a power system under steady-state operating conditions.[36]

It considers various factors, including:

- Generator real and reactive power injections
- Load real and reactive power demands
- Transmission line impedances
- Transformer tap settings

Key outputs of a power flow study include:

- **Voltage profile:** The magnitude and angle of voltage at each bus.
- **Real and reactive power flows** in each transmission line.
- **Power losses** in transmission lines.

To solve these equations, various numerical methods are employed, the most common of which include:

- **Newton-Raphson Method:** A second-order iterative technique that is widely used due to its fast convergence properties, particularly suitable for large systems.[37]
- **Gauss-Seidel Method:** An iterative method that is simpler but slower in convergence compared to the Newton-Raphson method, often used in smaller or less complex systems.[37]
- **Fast Decoupled Load Flow:** A simplification of the Newton-Raphson method that decouples the P and Q equations to speed up calculations.[37]

This analysis provides critical insights into voltage levels, power flows, and system losses throughout the network, facilitating optimal operational planning and the effective integration of new energy resources.[38]

Below figures are the voltage and reactive power values for each bus bar

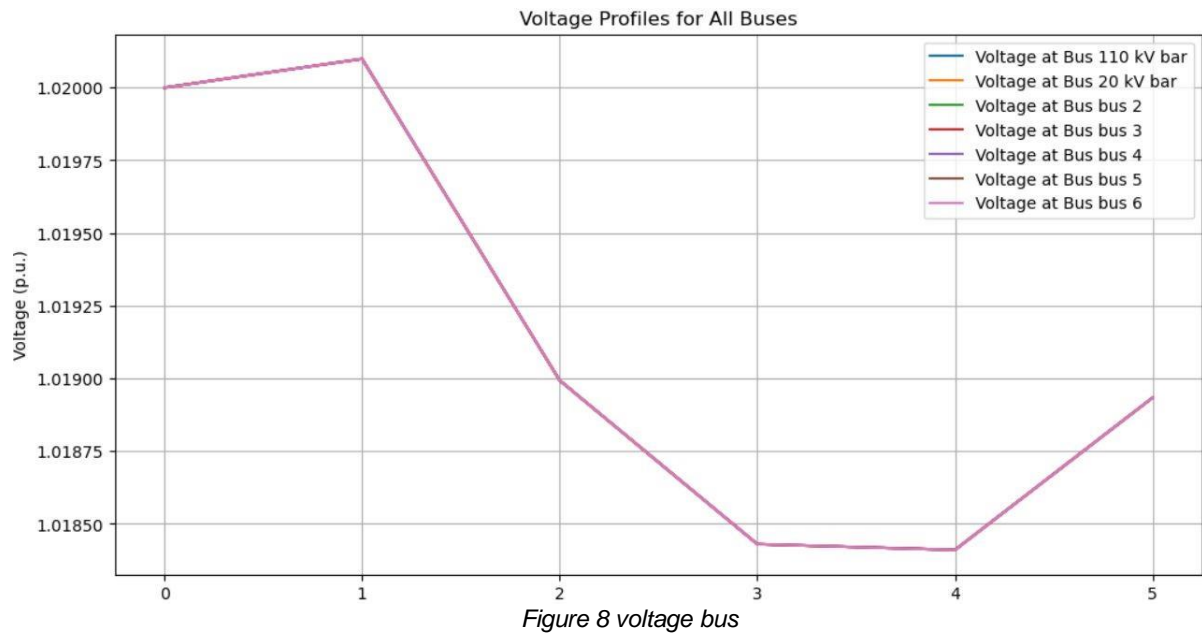


Figure 8 voltage bus

Figure 8 demonstrates the voltage profile for each bus bar. These values are from the data of the example network.

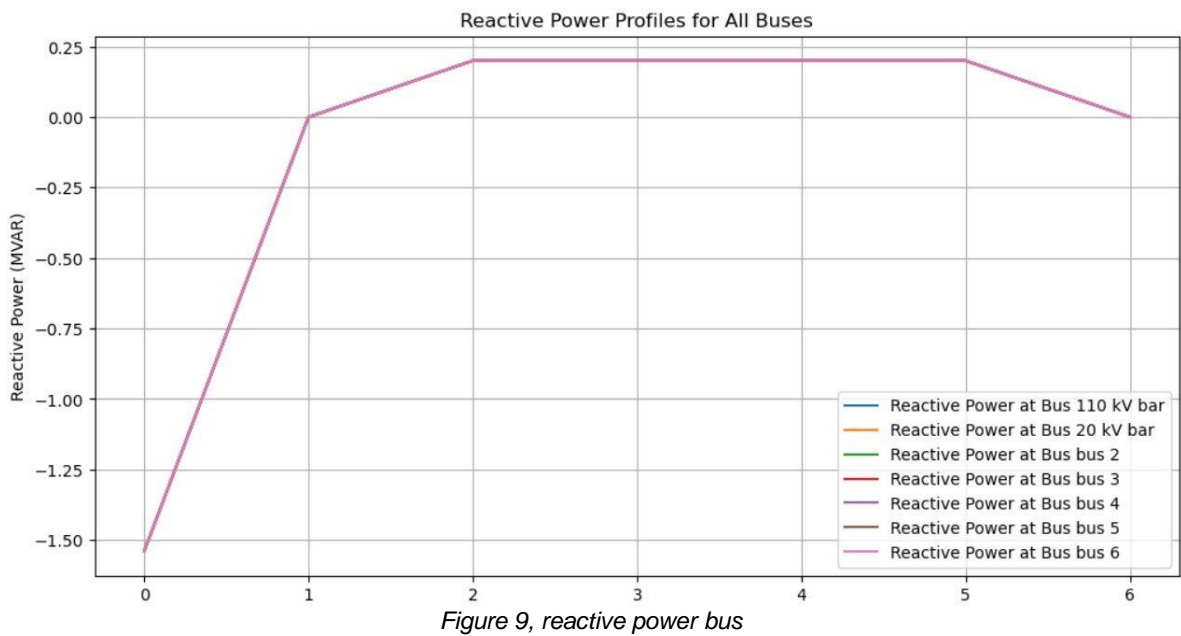


Figure 9, reactive power bus

Figure 9 shows the reactive power injected to each bus bar. Like the figure 8 these values are from the data of the example network

4.2 DATASET FOR MODELING

In order to test and train the model a set of data needed to be used. OPF is the tool here to generate the proper power where need to be injected to the buses. In the topology that has been used here include 4 buses (along with HV bus connect to the external grid)

Optimal Power Flow (OPF) is an advanced mathematical technique used in electrical power systems to determine the most economical operating conditions while satisfying system constraints like voltage limits, power flow on transmission lines, and generator output limits. The goal of OPF is to optimize a specific objective function, typically the cost of generation, while adhering to the physical and operational constraints of the network.[39, 40]

- **Objective Function:** Minimize the total generation cost, which is often a function of the power output of the generators.
- **Constraints:**
 - Power balance constraints: Ensure that the total generation meets the total demand plus system losses.
 - Generator limits: Each generator has a maximum and minimum output limit.
 - Voltage limits: Voltages at different buses must be kept within predefined bounds to ensure safe operation.
 - Transmission line limits: Power flowing through each transmission line must not exceed its thermal limit.

4.2.1 SAMPLE RESULTS

The data and code that has been used for OPF can be found in Appendix B

For each bus 288 samples have been captured. The following results have been generated for voltage and reactive power. The voltage profile for buses should be in acceptance range which is 1 [P.U] ± 0.05 .

The purpose of having this amount of data is to train and test the model.

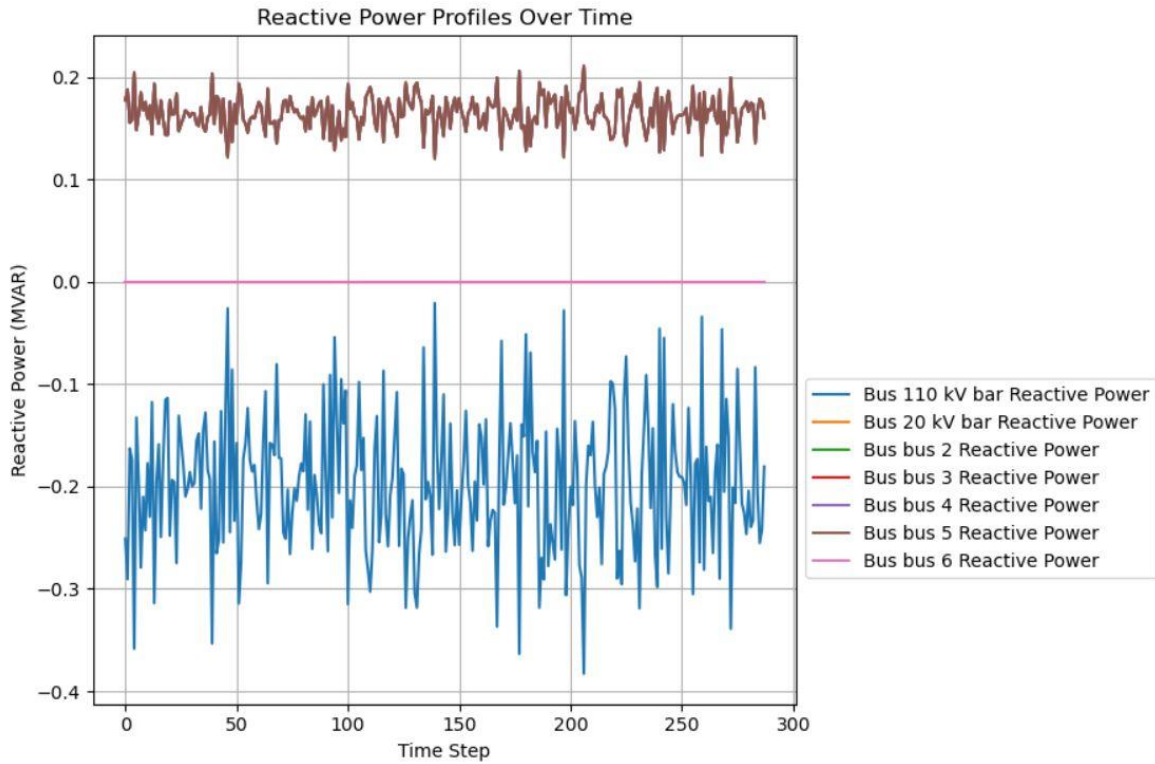


Figure 10, reactive power sample buses

This is the time series for reactive power. It can be observed that the DER reactive power penetration for buses 2, 3, 4, 5 are around the range 0.2 MVAR (all buses around same range), which is close to 0.2 MVAR for the mentioned buses.

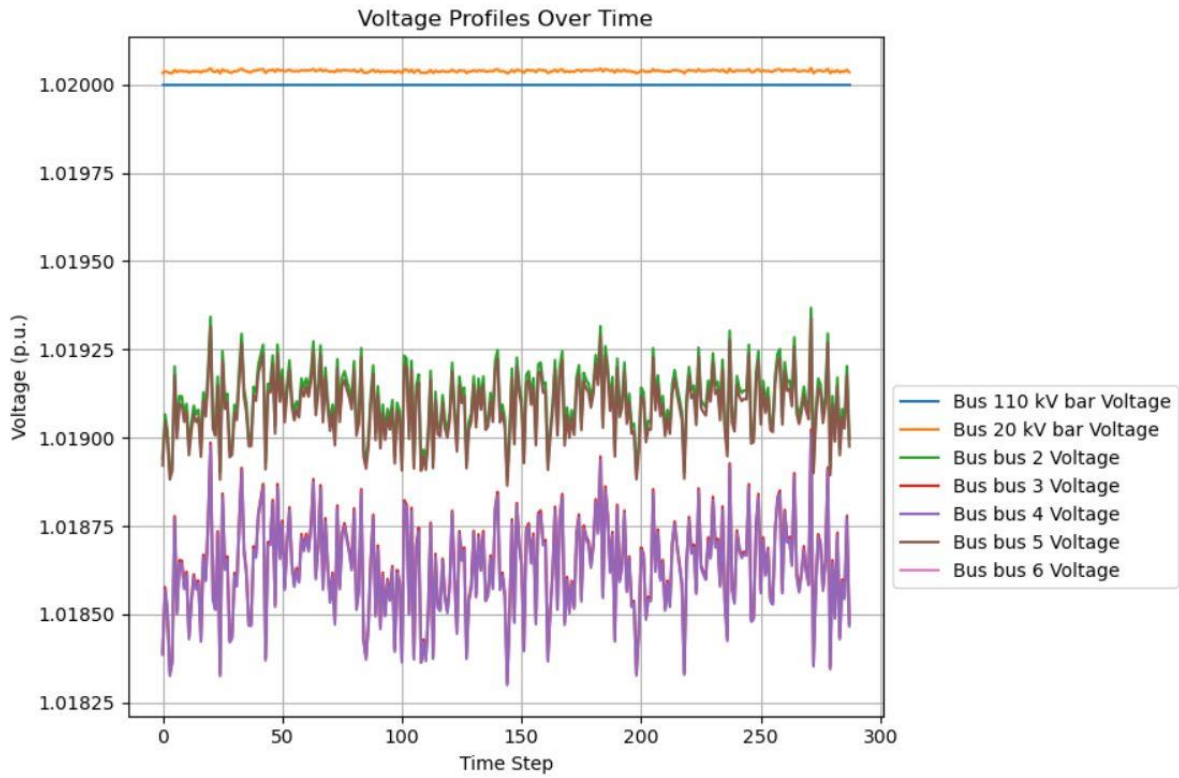


Figure 11, voltage sample buses

Voltage Profile for all buses are within the acceptance range, $1 \text{ [P.U]} \pm 0.05$. This is demonstration of voltage profile of buses that the DERs reactive power prediction will be conducted.

5 Results

In this section the results of SVM model will be mentioned.

5.1 SVM model

The dataset that has been collected previously will be used here to be used in SVM model.

In order to prepare the model half of the data have been used for train and the other ones have been used for testing the model. The code that used for this part is available in Appendix C

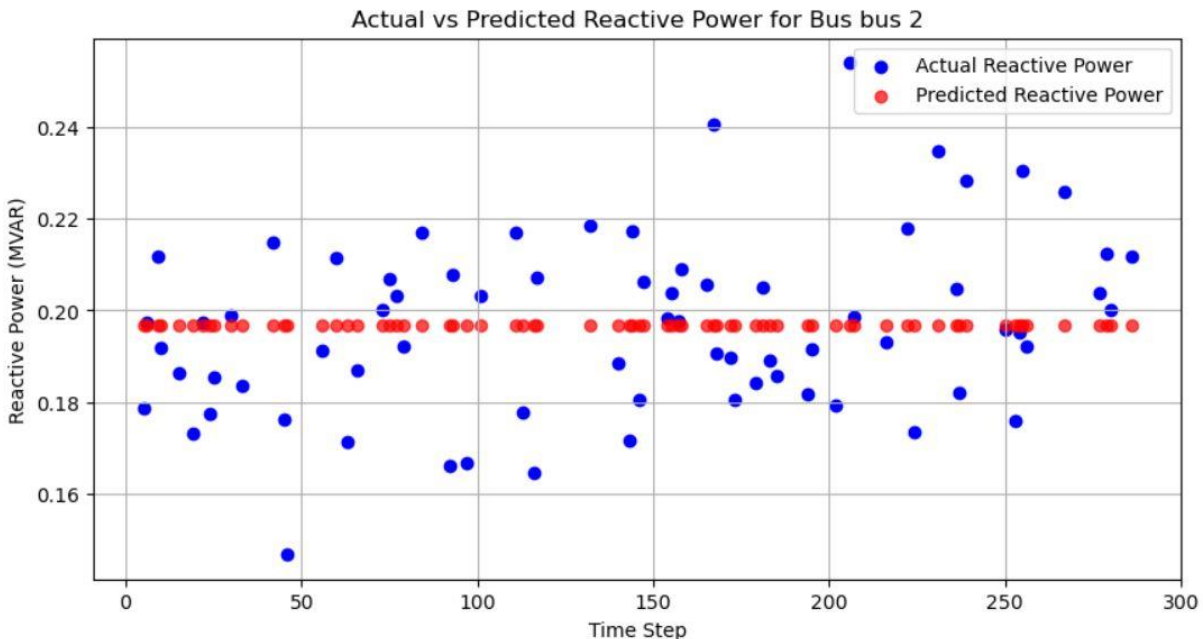


Figure 12, ML Predict Bus2

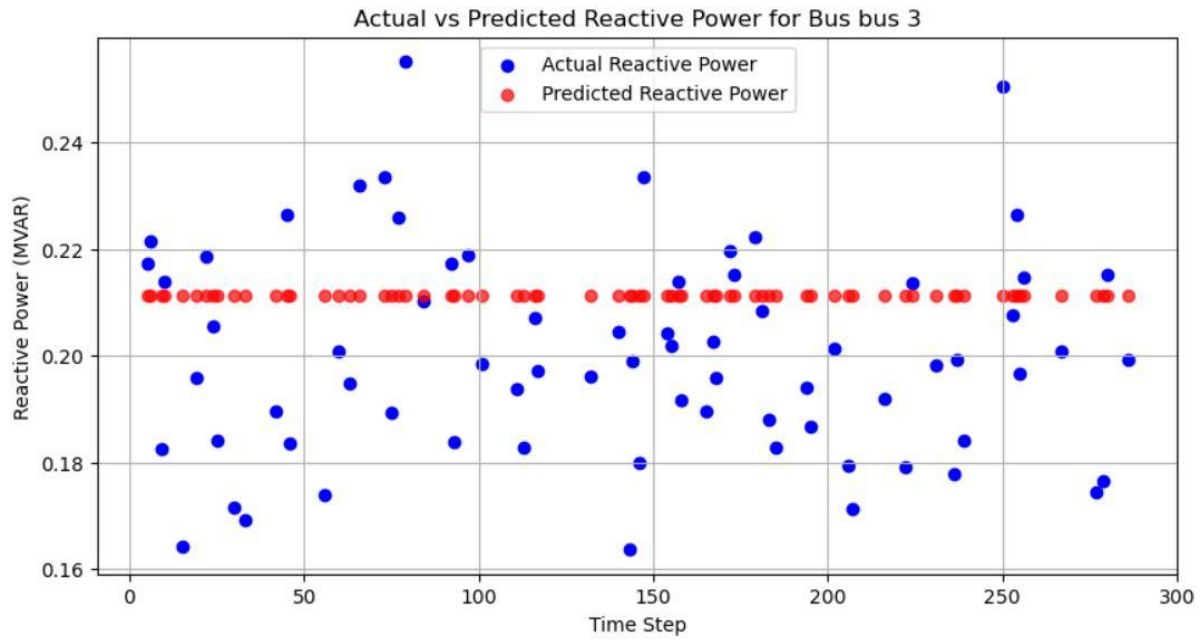


Figure 13, ML Predict Bus3

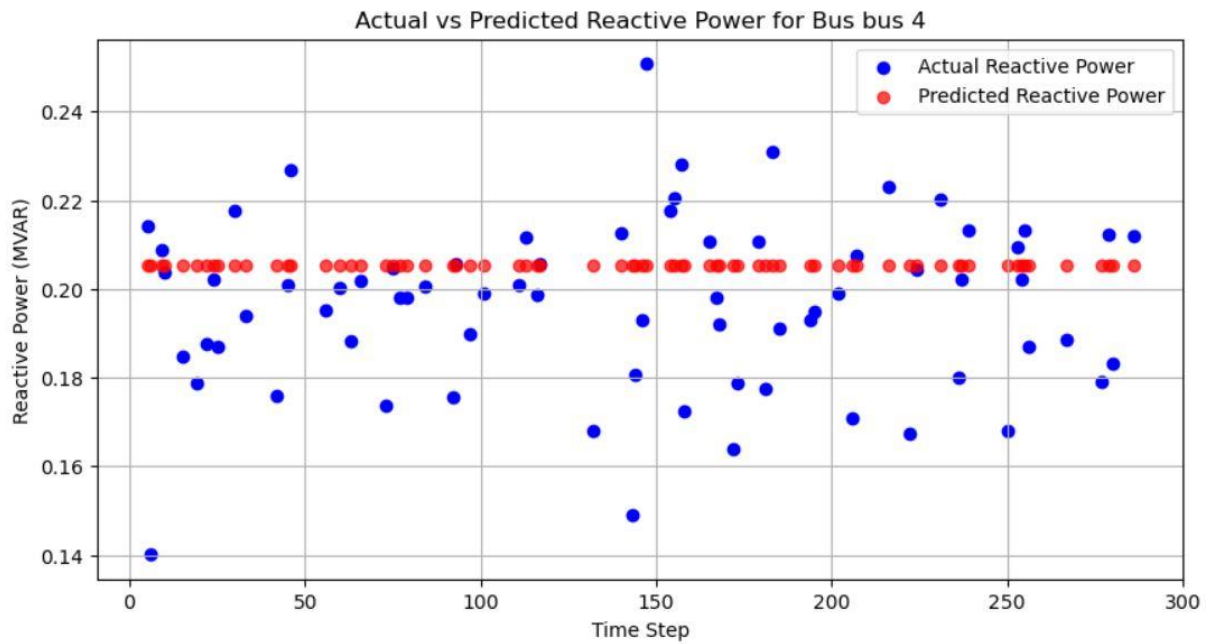


Figure 14, ML Predict Bus4

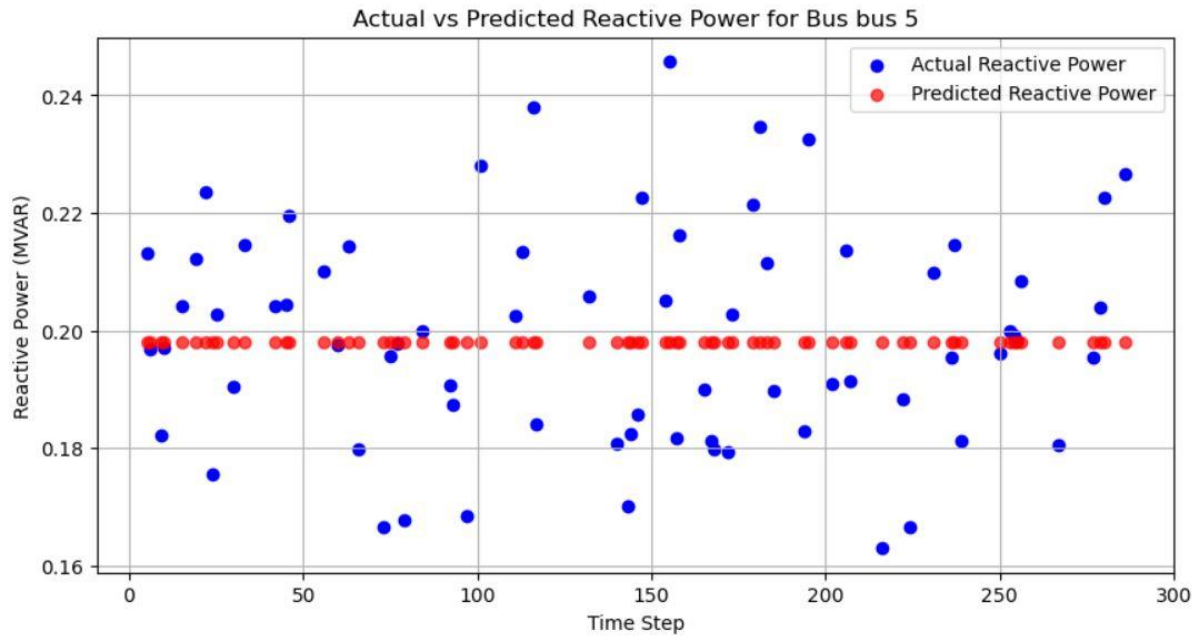


Figure 15, ML Predict Bus5

5.2 SVM Analysis

This section will be discussed regarding ML results for figures 12-15.

5.2.1 Underfitting and Overfitting

Evaluating underfitting and Overfitting is one of the main measures regarding model accuracy.

cross all buses, there's a tendency towards underfitting rather than overfitting. The models generally predict a smoothed version of the actual values, which indicates that they may not be capturing all the subtleties and dynamics of the reactive power fluctuations.

Buses 2 and 3, figures 12 & 13 show underfitting as the model predictions do not capture the full variability and peaks in the reactive power.

As for buses 4 and 5, figures 14 & 15, shows a balanced fit, tracking the actual data closely without sticking too tightly to the data.

5.2.2 Accuracy

accuracy varies with some buses showing high accuracy while others moderate.

For buses 2 and 3, figures 12 & 13 have moderate accuracy; predictions generally follow the trend of actual values but miss finer fluctuations and peaks, indicating a need for model adjustments to enhance sensitivity.

As for buses 4 and 5, figures 14 & 15 show high accuracy, with predictions closely aligning with actual values.

5.2.3 Robustness

The models show a degree of robustness in that they avoid large errors and capture the central trend reasonably well.

Buses 2, figures 12 and show good robustness in a general operational sense but lack the ability to adapt to sudden changes or capture extreme values effectively.

Buses 4 and 5, figures 14 & 15 demonstrate very good robustness, adapting well to the changes in reactive power over time and maintaining stability in predictions.

5.2.4 Performance

Performance is closely related to the accuracy and robustness. Buses with higher accuracy and robustness show better overall performance.

Regarding buses 2 and 3, figures 12 & 13 demonstrate adequate performance for general trends but struggle with the details and dynamics of the dataset. Their performance could be enhanced by increasing model sensitivity and possibly re-evaluating the feature set or preprocessing methods.

Buses 4 and 5, figures 14 & 15 are the better performers, showing good degree of accuracy and robustness which turn into superior overall performance. These ones replicate better the data and meet performance expectations.

6 Conclusion and Future Research

The analysis of the SVM models trained to predict reactive power for various buses using an RBF kernel reveals that while some buses exhibit high accuracy and performance, others show room for improvement, particularly in terms of capturing the reactive power's dynamics and fluctuations. Key findings include a tendency towards underfitting in several buses, suggesting that the model parameters need adjustment to better capture the peaks and troughs in the reactive power data.

Adjusting the gamma parameter in the SVM models could help improve sensitivity to fluctuations, potentially reducing underfitting and classing set of data could help to have more accurate results.

Adding more features or enhancing the preprocessing steps to better capture the dynamics and dependencies in the data might improve model performance, especially for buses 2 and 3, figures 12 & 13 where current models underperform.

In addition, having more real data such as weather conditions, past reactive power data, active power data, although can add complexity to the model however, it could lead to a more accurate prediction.

In addition, other ML Supervised Algorithms can be used to evaluate their capabilities including Recurrent Neural Networks (RNNs), Random Forest, Gradient Boosting Machines (GBMs).

Appendix A

Bus Number	DER Active Power (MW)	Reactive Power Capability (Mvar)
Bus 2	1.2	0.2
Bus 3	1.2	0.2
Bus 4	1.2	0.2
Bus 5	1.2	0.2

Power flow Analysis data & Code

```
import pandapower as pp
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.svm import SVR
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler

net = pp.create_empty_network()

# Adding buses
buses = ['110 kV bar', '20 kV bar', 'bus 2', 'bus 3', 'bus 4', 'bus 5', 'bus 6']
bus_ids = [pp.create_bus(net, name=bus, vn_kv=110 if '110' in bus else 20, type='b') for
bus in buses]

# External grid connection
pp.create_ext_grid(net, bus=bus_ids[0], vm_pu=1.02)

# Create an empty list for line IDs
line_ids = []

# Add lines
line_data = {'length_km': 1, 'std_type': "NAYY 4x150 SE"}
connections = [(1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 1)]
[line_ids.append(pp.create_line(net, from_bus=bus_ids[f-1], to_bus=bus_ids[t-1],
**line_data)) for f, t in connections]
# Add lines
line_data = {'length_km': 1, 'std_type': "NAYY 4x150 SE"}
connections = [(1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 1)]
[line_ids.append(pp.create_line(net, from_bus=bus_ids[f-1], to_bus=bus_ids[t-1],
**line_data)) for f, t in connections]

# Transformer
pp.create_transformer_from_parameters(net, hv_bus=bus_ids[0], lv_bus=bus_ids[1],
i0_percent=0.038, pfe_kw=11.6, vkr_percent=0.322,
sn_mva=40, vn_lv_kv=22, vn_hv_kv=110,
vk_percent=17.8)

# Add loads
loads = {'p_mw': 1, 'q_mvar': 0.2}
[pp.create_load(net, bus=bus_ids[i], **loads) for i in range(2, 7)]

# Run optimal power flow
pp.runopp(net, verbose=True)
```

```

bus_ids = net.bus.index

# Generate time series data for 288 time steps
n_timesteps = 288
time_steps = np.arange(n_timesteps)
load_variation = 0.1 # ±10% of the initial load

# Create time series for each load based on the OPF results
np.random.seed(0) # For reproducibility
load_profiles_p = {}
load_profiles_q = {}
for load in net.load.index:
    base_p = net.load.at[load, 'p_mw']
    base_q = net.load.at[load, 'q_mvar']
    load_profiles_p[load] = np.random.normal(base_p, base_p * load_variation, n_timesteps)
    load_profiles_q[load] = np.random.normal(base_q, base_q * load_variation, n_timesteps)

# Time series arrays for the simulation results
vm_pu_time_series = np.zeros((n_timesteps, len(bus_ids)))
q_mvar_time_series = np.zeros((n_timesteps, len(bus_ids)))

# Running OPF for each time step
for t in time_steps:
    for load in net.load.index:
        net.load.at[load, 'p_mw'] = load_profiles_p[load][t]
        net.load.at[load, 'q_mvar'] = load_profiles_q[load][t]
    pp.runopp(net)
    vm_pu_time_series[t, :] = net.res_bus.vm_pu.values
    q_mvar_time_series[t, :] = net.res_bus.q_mvar.values

# Data scaling and SVM training/testing
scaler = StandardScaler()

# Hyperparameter grid for tuning
param_grid = {
    'C': [0.1, 1, 10, 100],
    'gamma': [1e-4, 1e-3, 1e-2, 0.1, 1]
}

for idx, bus_id in enumerate(bus_ids):
    # Split and scale the data
    X_train, X_test, y_train, y_test = train_test_split(
        time_steps.reshape(-1, 1), q_mvar_time_series[:, idx], test_size=0.25,
        random_state=42)
    X_train_scaled = scaler.fit_transform(X_train)
    X_test_scaled = scaler.transform(X_test)

    # Perform Grid Search
    grid_search = GridSearchCV(SVR(kernel='rbf'), param_grid, cv=5, n_jobs=-1, verbose=2)
    grid_search.fit(X_train_scaled, y_train)

    # Train the SVM model with the best found parameters
    best_C = grid_search.best_params_['C']
    best_gamma = grid_search.best_params_['gamma']
    svm_regressor = SVR(kernel='rbf', C=best_C, gamma=best_gamma)
    svm_regressor.fit(X_train_scaled, y_train)

    # Predict the values
    y_pred = svm_regressor.predict(X_test_scaled)

    # Plot actual vs predicted values
    plt.figure(figsize=(10, 5))
    plt.scatter(X_test, y_test, color='blue', label='Actual Reactive Power')
    plt.scatter(X_test, y_pred, color='red', label='Predicted Reactive Power', alpha=0.7)
    plt.xlabel('Time Step')

```

```
plt.ylabel('Reactive Power (MVAR)')
plt.title(f'Actual vs Predicted Reactive Power for Bus {net.bus.at[bus_id, "name"]} -
Tuned Model')
plt.legend()
plt.grid(True)
plt.show()
```

Voltage & power demonstration for buses

```
import pandapower as pp
import pandapower.timeseries as timeseries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Create an empty network
net = pp.create_empty_network()

# Adding buses
buses = ['110 kV bar', '20 kV bar', 'bus 2', 'bus 3', 'bus 4', 'bus 5', 'bus 6']

bus_ids = [pp.create_bus(net, name=bus, vn_kv=110 if '110' in bus else 20, type='b') for bus
in buses]

# External grid connection
pp.create_ext_grid(net, bus=bus_ids[0], vm_pu=1.02)

# Initialize line_ids
```

```

line_ids = []

# Add lines

line_data = {'length_km': 1, 'std_type': "NAYY 4x150 SE"}

connections = [(1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 1)]

[line_ids.append(pp.create_line(net, from_bus=bus_ids[f-1], to_bus=bus_ids[t-1],
**line_data)) for f, t in connections]

# Transformer

pp.create_transformer_from_parameters(net, hv_bus=bus_ids[0], lv_bus=bus_ids[1],
                                     i0_percent=0.038, pfe_kw=11.6, vkr_percent=0.322,
                                     sn_mva=40, vn_lv_kv=22, vn_hv_kv=110, vk_percent=17.8)

# Add loads

loads = {'p_mw': 1, 'q_mvar': 0.2}

[pp.create_load(net, bus=bus_ids[i], **loads) for i in range(2, 7)]

# Run optimal power flow

pp.runopp(net, verbose=True)

# Define the time series for the loads

number_of_steps = 288 # For one day with 5-minute intervals

load_p_mw = np.random.normal(1, 0.1, number_of_steps) # Active power time series

load_q_mvar = np.random.normal(0.2, 0.05, number_of_steps) # Reactive power time series

# Create DataFrames for the time series data

```

```

df_load_p = pd.DataFrame({'load0_p': load_p_mw, 'load1_p': load_p_mw, 'load2_p':
load_p_mw, 'load3_p': load_p_mw, 'load4_p': load_p_mw})

df_load_q = pd.DataFrame({'load0_q': load_q_mvar, 'load1_q': load_q_mvar, 'load2_q':
load_q_mvar, 'load3_q': load_q_mvar, 'load4_q': load_q_mvar})

# Creating the data source

ds_load_p = pp.timeseries.DFData(df_load_p)

ds_load_q = pp.timeseries.DFData(df_load_q)

# Creating output writer to store results

ow = pp.timeseries.OutputWriter(net, time_steps=range(number_of_steps),
output_path='./time_series_results/', output_file_type=".csv")

ow.log_variable('res_bus', 'vm_pu')

ow.log_variable('res_bus', 'q_mvar')

# Running the time series simulation

pp.timeseries.run_timeseries(net, time_steps=range(number_of_steps))

# Plotting Voltage Profiles for All Buses

plt.figure(figsize=(12, 6))

for bus_id in bus_ids:

    plt.plot(net.res_bus.index, net.res_bus['vm_pu'], label=f'Voltage at Bus
{net.bus.name.at[bus_id]}')

plt.xlabel('Time Step')

plt.ylabel('Voltage (p.u.)')

plt.title('Voltage Profiles for All Buses')

plt.legend()

plt.grid(True)

plt.savefig('voltage_profiles.png')

```



```
plt.show()

# Plotting Reactive Power Profiles

plt.figure(figsize=(12, 6))

for bus_id in bus_ids:

    plt.plot(net.res_bus.index, net.res_bus['q_mvar'], label=f'Reactive Power at Bus
{net.bus.name.at[bus_id]}')

plt.xlabel('Time Step')

plt.ylabel('Reactive Power (MVAR)')

plt.title('Reactive Power Profiles for All Buses')

plt.legend()

plt.grid(True)

plt.savefig('reactive_power_profiles.png')

plt.show
```

Appendix B

Data Sampling Code

```
import pandapower as pp
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Generate time series data for 288 time steps
n_timesteps = 288
time_steps = np.arange(n_timesteps)

load_variation = 0.1

# Create time series for each load based on the load results from OPF
load_profiles_p = {}
load_profiles_q = {}
for load in net.load.index:
    base_p = net.load.at[load, 'p_mw']
    base_q = net.load.at[load, 'q_mvar']
```

```

np.random.seed(0)

load_profiles_p[load] = np.random.normal(base_p, base_p * load_variation, n_timesteps)

load_profiles_q[load] = np.random.normal(base_q, base_q * load_variation, n_timesteps)

# Apply these profiles to the loads and run a time simulation

# For simplicity, we will just recalculate OPF for each time step (not efficient for large
networks)

vm_pu_time_series = np.zeros((n_timesteps, len(bus_ids)))
q_mvar_time_series = np.zeros((n_timesteps, len(bus_ids)))

for t in time_steps:

    for load in net.load.index:

        net.load.at[load, 'p_mw'] = load_profiles_p[load][t]
        net.load.at[load, 'q_mvar'] = load_profiles_q[load][t]

    # Run OPF for the current time step

    pp.runopp(net)

    # Store results

    vm_pu_time_series[t, :] = net.res_bus.vm_pu.values
    q_mvar_time_series[t, :] = net.res_bus.q_mvar.values

print(net.res_bus.vm_pu.head())
print(net.res_bus.q_mvar.head())

# Plot results

# Voltage Profile Plot

plt.figure(figsize=(12, 6))

for idx, bus_id in enumerate(bus_ids):

```

```

plt.plot(time_steps, vm_pu_time_series[:, idx], label=f'Bus {net.bus.name.at[bus_id]}
Voltage')

plt.xlabel('Time Step')

plt.ylabel('Voltage (p.u.)')

plt.title('Voltage Profiles Over Time')

plt.legend(loc='upper left', bbox_to_anchor=(1, 0.5)) # Position the legend outside the plot

plt.grid(True)

plt.tight_layout(rect=[0, 0, 0.75, 1]) # Adjust the plot area to make room for the legend

plt.savefig('time_series_voltage_profile.png')

plt.show()

```

Reactive Power Profile Plot

```

plt.figure(figsize=(12, 6))

for idx, bus_id in enumerate(bus_ids):

    plt.plot(time_steps, q_mvar_time_series[:, idx], label=f'Bus {net.bus.name.at[bus_id]}
Reactive Power')

plt.xlabel('Time Step')

plt.ylabel('Reactive Power (MVAR)')

plt.title('Reactive Power Profiles Over Time')

plt.legend(loc='upper left', bbox_to_anchor=(1, 0.5)) # Position the legend outside the plot

plt.grid(True)

plt.tight_layout(rect=[0, 0, 0.75, 1]) # Adjust the plot area to make room for the legend

plt.savefig('time_series_reactive_power_profile.png')

plt.show()

```

Appendix C

SVM Code

```
import pandapower as pp
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.svm import SVR
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler

# Create an empty network
net = pp.create_empty_network()

# Adding buses
buses = ['110 kV bar', '20 kV bar', 'bus 2', 'bus 3', 'bus 4', 'bus 5', 'bus 6']
bus_ids = [pp.create_bus(net, name=bus, vn_kv=110 if '110' in bus else 20, type='b') for bus
in buses]

# External grid connection
```

```

pp.create_ext_grid(net, bus=bus_ids[0], vm_pu=1.02)

# Create an empty list for line IDs
line_ids = []

# Add lines
line_data = {'length_km': 1, 'std_type': "NAYY 4x150 SE"}
connections = [(1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 1)]
[line_ids.append(pp.create_line(net, from_bus=bus_ids[f-1], to_bus=bus_ids[t-1],
**line_data)) for f, t in connections]

# Add lines
line_data = {'length_km': 1, 'std_type': "NAYY 4x150 SE"}
connections = [(1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 1)]
[line_ids.append(pp.create_line(net, from_bus=bus_ids[f-1], to_bus=bus_ids[t-1],
**line_data)) for f, t in connections]

# Transformer
pp.create_transformer_from_parameters(net, hv_bus=bus_ids[0], lv_bus=bus_ids[1],
i0_percent=0.038, pfe_kw=11.6, vkr_percent=0.322,
sn_mva=40, vn_lv_kv=22, vn_hv_kv=110, vk_percent=17.8)

# Add loads
loads = {'p_mw': 1, 'q_mvar': 0.2}
[pp.create_load(net, bus=bus_ids[i], **loads) for i in range(2, 7)]

# Run optimal power flow
pp.runopp(net, verbose=True)

```

```

bus_ids = net.bus.index

# Generate time series data for 288 time steps
n_timesteps = 288
time_steps = np.arange(n_timesteps)
load_variation = 0.1 # ±10% of the initial load

# Create time series for each load based on the OPF results
np.random.seed(0) # For reproducibility
load_profiles_p = {}
load_profiles_q = {}
for load in net.load.index:
    base_p = net.load.at[load, 'p_mw']
    base_q = net.load.at[load, 'q_mvar']
    load_profiles_p[load] = np.random.normal(base_p, base_p * load_variation, n_timesteps)
    load_profiles_q[load] = np.random.normal(base_q, base_q * load_variation, n_timesteps)

# Time series arrays for the simulation results
vm_pu_time_series = np.zeros((n_timesteps, len(bus_ids)))
q_mvar_time_series = np.zeros((n_timesteps, len(bus_ids)))

# Running OPF for each time step
for t in time_steps:
    for load in net.load.index:
        net.load.at[load, 'p_mw'] = load_profiles_p[load][t]
        net.load.at[load, 'q_mvar'] = load_profiles_q[load][t]

```

```

pp.runopp(net)

vm_pu_time_series[t, :] = net.res_bus.vm_pu.values
q_mvar_time_series[t, :] = net.res_bus.q_mvar.values

# Data scaling and SVM training/testing
scaler = StandardScaler()

# Hyperparameter grid for tuning
param_grid = {
    'C': [0.1, 1, 10, 100],
    'gamma': [1e-4, 1e-3, 1e-2, 0.1, 1]
}

for idx, bus_id in enumerate(bus_ids):
    # Split and scale the data
    X_train, X_test, y_train, y_test = train_test_split(
        time_steps.reshape(-1, 1), q_mvar_time_series[:, idx], test_size=0.25, random_state=42)
    X_train_scaled = scaler.fit_transform(X_train)
    X_test_scaled = scaler.transform(X_test)

    grid_search = GridSearchCV(SVR(kernel='rbf'), param_grid, cv=5, n_jobs=-1, verbose=2)
    grid_search.fit(X_train_scaled, y_train)

# Train the SVM model
best_C = grid_search.best_params_['C']
best_gamma = grid_search.best_params_['gamma']

```



```
svm_regressor = SVR(kernel='rbf', C=best_C, gamma=best_gamma)
svm_regressor.fit(X_train_scaled, y_train)

# Predict the values
y_pred = svm_regressor.predict(X_test_scaled)

# Plot actual vs predicted values
plt.figure(figsize=(10, 5))
plt.scatter(X_test, y_test, color='blue', label='Actual Reactive Power')
plt.scatter(X_test, y_pred, color='red', label='Predicted Reactive Power', alpha=0.7)
plt.xlabel('Time Step')
plt.ylabel('Reactive Power (MVAR)')
plt.title(f'Actual vs Predicted Reactive Power for Bus {net.bus.at[bus_id, "name"]} - Tuned Model')
plt.legend()
plt.grid(True)
plt.show()
```

References

[1]: State-of-the-art of hosting capacity in modern power systems with distributed generation
Sherif M. Ismael a , Shady H.E. Abdel Aleem b , Almoataz Y. Abdelaziz c , Ahmed F. Zobaa
d

[2]: P.N. Vovos, A.E. Kiprakis, A.R. Wallace, G.P. Harrison, Centralized and distributed
voltage control: impact on distributed generation penetration, IEEE Trans. Power Syst. 22
(2007) 476e483,

[3]: M. Ebad, W.M. Grady, An approach for assessing high-penetration PV impact on
distribution feeders, Elec. Power Syst. Res. 133 (2016) 347e354

[4]: Power system and market integration of DER, The EU-DEEP approach, J. Deuse; D.
Benintendi; P. J. Agrell; P. Bogetoft, CIRED 2005 - 18th International Conference and
Exhibition on Electricity Distribution

[5]: R.A. Shayani, M.A.G. De Oliveira, Photovoltaic generation penetration limits in radial
distribution systems, IEEE Trans. Power Syst. 26 (2011) 1625e1631,

[6]: Increasing the hosting Capacity of Radial Distribution Grids in Jordan Fadi AlAlamat

- [7]: High-Penetration PV Integration Handbook for Distribution Engineers Rich Seguin, Jeremy Woyak, David Costyk, and Josh Hambrick Electrical Distribution Design Barry Mather National Renewable Energy Laboratory
- [8]: Increasing the photovoltaic-system hosting capacity of low voltage distribution networks, T Degner, G Arnold, T Reimann, B Engel, M Breede, 2011
- [9]: M. Bollen, F. Hassan, Integration of Distributed Generation in the Power System, Wiley-IEEE Press, Hoboken, USA, 2011
- [10]: Increasing the photovoltaic-system hosting capacity of low voltage distribution networks, T Degner, G Arnold, T Reimann, B Engel, M Breede, 2011
- [11]: M. Bollen, F. Hassan, Integration of Distributed Generation in the Power System, Wiley- IEEE Press, Hoboken, USA, 2011.
- [12]: J. Deuse, S. Grenard, M.H.J. Bollen, M. Héger, F. Sollerkvist, Effective impact of DER on distribution system protection, Conf. Proc. 19th Int. Conf. Exhib. Electr. Distrib. (2007).
- [13]: M. Vandenberg, R. Hermes, V. Helmbrecht, H. Loew, D. Craciun, Technical solutions supporting the large-scale integration of photovoltaic systems in the future distribution grids, in: 22nd Int. Conf. Exhib. Electr. Distrib., 2013.
- [14]: Voltage Profile of Hosting Capacity Enhancement Based on Smart Inverter Reactive Power Control for PV Grid Connected System Ahmed Alfouly¹, I. Hamdan¹, Mohamed A. Ismeil¹, 2023
- [15]: Power System Harmonics and Passive Filter Designs, 2015.
- [16] Y. Li, "Deep reinforcement learning: An overview," arXiv Prepr. arXiv1701.07274, 2017.
- [17] J. L. Rojo-Álvarez, M. Martínez-Ramón, J. M. Marí, and G. Camps-Valls, Digital signal processing with Kernel methods. Wiley Online Library, 2018.
- [18] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. MIT press, 2018.

- [19] CIGRE Task Force C6.04.02 (2011) Benchmark Systems for Network Integration of Renewable Energy Resources, Version 7
- [20] Chakrabarti, Bijaya Kumar, and Lakshmi S. Kumar. "Short-term reactive power forecasting using support vector machines." *IEEE Transactions on Power Systems* 20.1 (2005): 456-462.
- [21] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." *Machine learning* 20.3 (1995): 273-297.
- [22] James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. "An introduction to statistical learning." Springer (2013).
- [23] Kavousi-Hakami, Mojtaba, et al. "Day-ahead load forecasting using a novel ensemble of ANFIS and SVM." *IEEE Transactions on Smart Grid* 3.4 (2012): 1630-1638.
- [24] Hassan, Mahmoud Y., et al. "Support vector machine for short-term load forecasting." International.
- [25] Zhao, J., et al. "A hybrid forecasting method for short-term wind power based on SVR and improved PSO." 2009 International Conference on Sustainable Power Generation and Supply. IEEE, 2009.
- [26] Cristianini, Nello, and John Shawe-Taylor. "An introduction to support vector machines and other kernel-based learning methods." Cambridge university press, 2000.
- [27] James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. "An introduction to statistical learning." Springer (2013).
- [28] Hsu, Chih-Wei, Chih-Chung Chang, and Chih-Jen Lin. "A practical guide to support vector classification." (2001)
- [29] Chakrabarti, Bijaya Kumar, and Lakshmi S. Kumar. "Short-term reactive power forecasting using support vector machines." *IEEE Transactions on Power Systems* 20.1 (2005)

- [30] Zhao, J., et al. "A hybrid forecasting method for short-term wind power based on SVR and improved PSO." 2009 International Conference on Sustainable Power Generation and Supply. IEEE, 2009
- [31] Hsu, Chih-Wei, Chih-Chung Chang, and Chih-Jen Lin. "A practical guide to support vector classification (2001)
- [32] Cristianini, Nello, and John Shawe-Taylor. "An introduction to support vector machines and other kernel-based learning methods." Cambridge university press, 2000. [invalid URL removed] (This textbook chapter provides a detailed explanation of RBF kernels, gamma, and sigma)
- [33] Bergstra, James, and Yoshua Bengio. "Random search for hyper-parameter optimization." Journal of Machine Learning Research 13.feb (2012)
- [34] Hsu, Chih-Wei, Chih-Chung Chang, and Chih-Jen Lin. "A practical guide to support vector classification." (2001)
- [35] James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. "An introduction to statistical learning with applications in R." Springer New York, NY, 2013
- [36] Glover, J. D., Sarma, M. S., & Overbye, T. J. (2012). *Power system analysis and design*. SI edition (Vol. 1). Cengage Learning.
- [37] K. Uma Rao, "Power System Operation and Control," ISBN 8120331918, PHI Learning, 2009
- [38] Paul M. Anderson, "Analysis of Faulted Power Systems," ISBN 0780311450, IEEE Press, 1995.
- [39] J. Duncan Glover, Mulukutla S. Sarma, Thomas Overbye, "Power System Analysis & Design," ISBN 1111425779, Cengage Learning, 2012.
- [40] Grainger & Stevenson, "Power System Analysis," ISBN 0070612935, McGraw-Hill Education, 1994.
- [41] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*. Springer Science & Business Media.

[42] Smart Distribution Network Situation Awareness for High-Quality Operation and Maintenance: A Brief Review MDPI

[43] Smart Distribution Networks: A Review of Modern Distribution Concepts from a Planning Perspective MDPI

[44] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT press, 2016.

[45] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018

[46] Graves, Alex. "Long short-term memory networks learn problems." Proceedings of the 2014 IEEE symposium on computational intelligence and cognitive algorithms (CICA). IEEE, 2014

